



# Cormas, une plate-forme multi-agent pour concevoir collectivement des modèles et interagir avec les simulations

Nicolas Becu, Pierre Bommel, Christophe Le Page, François Bousquet

## ► To cite this version:

Nicolas Becu, Pierre Bommel, Christophe Le Page, François Bousquet. Cormas, une plate-forme multi-agent pour concevoir collectivement des modèles et interagir avec les simulations. Journées Francophones sur les Systèmes Multi-Agents (JFSMA), Fabien MICHEL, Julien SAUNIER, Oct 2016, Saint-Martin du Vivier (Rouen), France. hal-01385722

**HAL Id: hal-01385722**

**<https://hal.science/hal-01385722>**

Submitted on 22 Oct 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Cormas, une plate-forme multi-agent pour concevoir collectivement des modèles et interagir avec les simulations

Nicolas Becu<sup>a</sup>  
nicolas.becu@univ-lr.fr

Pierre Bommel<sup>b,c</sup>  
bommel@cirad.fr

Christophe Le Page<sup>b,d</sup>  
le\_page@cirad.fr

François Bousquet<sup>b</sup>  
bousquet@cirad.fr

<sup>a</sup> CNRS - UMR 7266 LIENSs, La Rochelle, France

<sup>b</sup> CIRAD - UPR GREEN, Montpellier, France

<sup>c</sup> Universidad de Costa Rica, CIEDA, Costa Rica

<sup>d</sup> Universidade de Brasilia, CDS, Brésil

## Résumé

Ce document présente les nouvelles fonctionnalités de Cormas, une plate-forme de modélisation multi-agent dédiée à la gestion des ressources renouvelables. Logiciel libre, Cormas vise à concevoir facilement un SMA et analyser des scénarios. Aujourd'hui, Cormas a pris une direction novatrice orientée vers la conception collective de modèles et la simulation interactive. Ces simulations hybrides mixent des décisions prises par les joueurs et d'autres par le modèle. Cela permet d'interagir avec une simulation en modifiant le comportement des agents et la façon dont ils utilisent les ressources. Ainsi, il est possible d'explorer collectivement des scénarios à moyen et long terme afin de mieux comprendre comment atteindre une situation souhaitée. En retour, cela permet de réviser collectivement le modèle conceptuel. Après avoir expliqué la philosophie de la modélisation d'accompagnement, ce document présente la façon dont les fonctionnalités de Cormas sont appliquées à travers trois expériences.

**Mots-clés :** modélisation participative, simulation interactive, gestion des ressources naturelles, implication des acteurs, système multi-agent

## Abstract

*This paper presents the new functionalities of Cormas, a multi-agent modeling platform dedicated to the management of renewable resources. As free software, Cormas is intended to facilitate the design of ABM as well as the monitoring and analysis of simulation scenarios. Today Cormas has*

*taken an innovative direction oriented towards the collective design of models and interactive simulation. These hybrid simulations are mixing decisions taken by stakeholders and others by the model. This allows the user to interact with a simulation by changing the behavior of agents. Thus, it is possible to collectively explore medium and long-term scenarios to better understand how a desired situation may be reached. In turn, this feed back into the collective design of the model. After having explained the philosophy of the companion modeling, this paper presents how the Cormas functionalities are put into practice through three experiments with stakeholders facing actual environmental challenges.*

## 1 Introduction

Cormas est une plate-forme de modélisation multi-agent dédiée à la gestion des ressources renouvelables [1]. En tant que logiciel libre, il est utilisé par une communauté internationale de chercheurs [2] étudiant les relations entre les sociétés et leur environnement. Cormas est destiné à faciliter la conception de SMA, ainsi que le suivi et l'analyse des simulations. En effet, le but d'un SMA est de comprendre comment des entités indépendantes peuvent interagir, se coordonner et co-évoluer, tout en produisant des effets sur le système. Un agent peut être décrit comme une entité autonome capable de s'adapter à un environnement changeant. Réciproquement ses actions peuvent modifier cet environnement. Etant centré sur l'individu, un SMA permet à l'utilisateur d'une simulation d'assumer le rôle d'un agent et ainsi de « penser comme un

loup, un mouton ou une mouche » [3].

Ces dernières années, le développement de Cormas a pris une direction innovante orientée vers la modélisation participative, à savoir la conception collective de modèles et la simulation interactive. Cette orientation répond à la demande croissante de notre communauté de pratique. Parallèlement à son développement, une méthodologie de modélisation appelée approche ComMod (pour Companion-Modeling, [4], [5]) a été mise en place et formalisée. Elle suppose que les acteurs du développement peuvent *décider* de leurs objectifs à long terme sur la base d'une conception partagée de la situation actuelle. Il est alors utile d'explorer des scénarios collectivement pour mieux comprendre si une situation souhaitée peut être atteinte. Le modèle sous-jacent de ces simulations dépend de la manière dont les acteurs sont représentés. On distingue deux types de représentation : (i) des agents virtuels exécutant des activités prédéfinies dans un SMA informatisé, ou (ii) des acteurs humains jouant leur rôle dans un jeu de rôle (JdR). Même si ce n'est pas dans le sens classique du terme, un JdR peut être vu comme une représentation du monde, à savoir un modèle. Mais entre ces deux extrêmes, des situations intermédiaires existent où certaines décisions sont faites par des joueurs et d'autres sont automatisées par l'ordinateur. Le terme « modèle de simulation hybride » couvre ces situations intermédiaires [6].

Dans ce but, nous développons Cormas pour (a) faciliter la conception collective et l'implémentation de SMA, et (b) permettre le développement de simulations interactives, afin que les acteurs puissent participer activement, seuls ou avec d'autres, dans le déroulement d'un scénario.

Si la gestion adaptative et la participation du public sont devenues des mots à la mode, dans la pratique ces termes sont souvent juste des expressions galvaudées, utilisées par les scientifiques pour justifier l'extraction d'information sans réelle participation des acteurs concernés [7]. La modélisation participative devrait promouvoir la reconnaissance mutuelle des perceptions, l'appropriation des connaissances et la prise de décision collective. Il est donc nécessaire de disposer d'outils facilement modifiables pour agir sur la simulation et modifier le modèle conceptuel. Nos différentes

expériences sur le terrain ont montré la nécessité de créer une continuité entre le modèle conceptuel et son implémentation. Cette nouvelle version contribue à atteindre cet objectif.

## 2 Aperçu de Cormas

En tant que *framework* qui propose des classes prédéfinies et un ensemble d'outils de visualisation, Cormas est destiné à faciliter l'implémentation de SMA, ainsi que le suivi et l'analyse des simulations. Il utilise VisualWorks [8], un environnement de programmation basé sur Smalltalk, un langage purement orienté objet. Un *framework* signifie que le modélisateur implémente son modèle en spécialisant des classes prédéfinies. Elles contiennent des attributs et des méthodes génériques qui peuvent être réutilisés. Principalement, trois types d'entités génériques sont disponibles : «agent social», «entité spatiale» et «entité passive». Ainsi, pour créer un Ruminant par exemple, le modélisateur peut le définir comme un sous-type d'*AgentLocation* (agent social situé). Il sera alors en mesure de se déplacer sur l'espace ou de percevoir ses voisins en appelant les méthodes génériques (*#moveTo*: et *#perceive*) définies au niveau de *AgentLocation*.

*CormasModel* est l'ordonnanceur abstrait du modèle. Il doit gérer la dynamique globale de la simulation. Chaque modèle créé sous Cormas spécialise cette classe qui peut alors réutiliser de nombreuses méthodes prédéfinies pour instancier l'état initial d'une simulation et pour activer les entités. Au cours d'une simulation, toute nouvelle instance d'agent est collectée par l'ordonnanceur qui dispose de listes toujours à jour des entités (en enlevant les agents morts et en ajoutant les nouveaux).

L'avantage d'utiliser une plate-forme est également qu'il libère le modélisateur de nombreuses contraintes de codage. Comme Cormas est conforme à l'architecture MVC, il permet de se concentrer uniquement sur son sujet sans se soucier des accessoires qui viennent avec un simulateur. Après avoir codé les agents et autres entités, le modélisateur doit simplement indiquer la façon dont celles-ci sont activées par l'ordonnanceur. Enfin, il peut spécifier la façon dont il veut visualiser les entités et les sorties de son modèle. Pour cette phase, plusieurs interfaces sont

disponibles qui évitent de coder l'affichage du modèle et les courbes des variables. Une grille spatiale permet de montrer le paysage virtuel et les agents. Diverses façons de les afficher peuvent être sélectionnées afin de voir le SMA selon différents "points de vue" (PoV).

### **3 Modélisation participative en appui à l'approche ComMod**

La nouvelle orientation vise à répondre à la demande croissante de notre communauté de pratique vers plus de participation en favorisant la conception collective et la simulation interactive de SMA.

En parallèle avec le développement de Cormas, l'approche ComMod [5], [11], [13] a été définie et appliquée dans de nombreux pays. *Companion* signifie que le but est d'aider les acteurs du développement à définir leurs propres objectifs à long terme et à les «accompagner» au lieu de proposer une formule clé en main pour la gestion des ressources [14]. Dans ce domaine, il semble nécessaire de prendre une certaine distance avec la posture positiviste qui conçoit les connaissances scientifiques comme seules valables. A l'instar de l'épistémologie constructiviste (dans le sens de [15] et [16]), l'approche ComMod cherche à «construire» collectivement des connaissances sur la base des perceptions des parties prenantes et de leurs expériences sociales. Au-delà des systèmes traditionnels d'aide à la décision, cette approche participative aspire à la construction d'une représentation partagée basée sur la façon dont les participants perçoivent la situation actuelle et la façon dont elle pourrait ou devrait évoluer. En tant qu'approche de médiation, ComMod présuppose que les acteurs doivent être bien informés des enjeux et qu'ils ont tous un intérêt à résoudre le problème d'origine.

Pour mieux comprendre une situation donnée, la conception collective d'un SMA est effectuée afin de garantir la reconnaissance de cette représentation par l'ensemble des participants. Ainsi, la modélisation se rapproche plus d'une plate-forme de communication pour faciliter l'apprentissage collectif qu'un itinéraire prédéfini pour le pilotage des ressources renouvelables [17]. Pour faciliter cette phase difficile, l'utilisation

de jeux de rôle (JdR) est souvent proposée. Les participants y jouent leur propre rôle dans une situation virtuelle simplifiée. Or ce jeu est déjà une représentation du monde, donc un modèle [18]. En jouant le rôle qui représente son activité dans la vie réelle, un joueur fournit des informations pour le modélisateur [19], [20]. Cette approche est bien reconnue dans les SMA dits empiriques [21] : les règles, les comportements et les décisions prises au cours du jeu par les joueurs sont utilisés pour spécifier les agents informatiques [22]. Les discussions lors du débriefing permettent de confirmer ou de réviser certaines parties de ce modèle conceptuel.

A partir du modèle conceptuel, la mise en place d'un SMA offre la possibilité d'explorer des scénarios. Car concevoir un SMA ne donne pas immédiatement accès à la compréhension de son comportement. Le temps joue un rôle actif et décisif en activant les entités. La séquence des activités et des interactions produit des résultats difficilement prévisibles. Même si les mécanismes élémentaires sont simples, nous ne sommes pas en mesure de prendre en compte de nombreux éléments qui s'influencent les uns les autres [23]. C'est cette mise en mouvement, cette animation (du latin *animare* « donner vie ») qui fait s'exprimer le modèle.

Ainsi, la simulation permet de comprendre le fonctionnement du SMA et d'évaluer si l'évolution du système virtuel est cohérente avec celui qu'il est censé imiter. En confrontant la perception commune du système réel avec son évolution virtuelle, la simulation permet de corriger le modèle conceptuel afin d'être plus conforme à la situation observée.

### **4 Les nouvelles fonctionnalités de Cormas, orientées vers la modélisation participative**

Au lieu de regarder une simulation sans interférer avec le processus, une simulation interactive vise à évaluer les différentes décisions prises par les agents. Pour cela, les participants peuvent modifier les paramètres ou peuvent envoyer des ordres spécifiques ou même modifier la stratégie principale d'un agent. En interagissant avec le système virtuel par le biais de son *avatar*, les participants

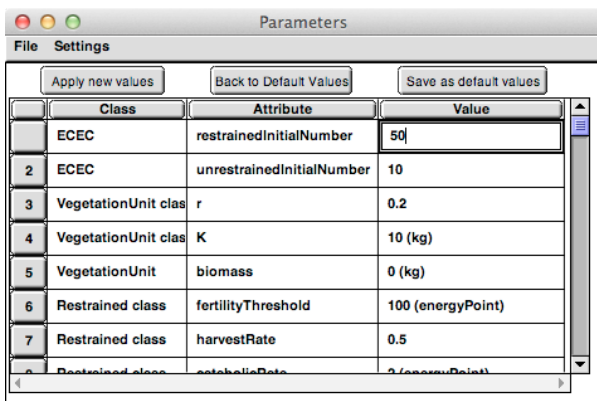


peuvent tester des stratégies alternatives ou de nouvelles pratiques afin d'en évaluer les conséquences.

Les développements récents de Cormas visent à proposer plusieurs outils d'interaction pour faciliter cette approche prospective participative. Ils permettent aux participants de contribuer activement à la conception d'un SMA et d'interagir avec le simulateur. Ces outils ont été élaborés à partir d'expériences ComMod concrètes. D'autres développements sont encore en cours pour répondre aux besoins croissants des études de cas récentes.

#### 4.1 Concevoir un modèle

Comme Cormas est un *framework*, le modélisateur doit spécialiser certaines classes prédéfinies, principalement des entités «sociales», «spatiales» ou «passives». Lors de la définition d'une classe, des attributs peuvent être ajoutés. Cormas aide le modélisateur dans le réglage de la valeur initiale de ces attributs. Ainsi la valeur initiale de l'attribut «energy» de la classe *Ruminant* peut être fixée à 50 points par défaut. Cela signifie que lors de l'initialisation d'une simulation, tous les ruminants commencent avec 50 points d'énergie. Et pendant la simulation, chaque nouvelle instance commencera aussi avec 50 points.



	Class	Attribute	Value
	ECEC	restrainedInitialNumber	50
2	ECEC	unrestrainedInitialNumber	10
3	VegetationUnit class	r	0.2
4	VegetationUnit class	K	10 (kg)
5	VegetationUnit	biomass	0 (kg)
6	Restrained class	fertilityThreshold	100 (energyPoint)
7	Restrained class	harvestRate	0.5
8	Restrained class	catchallRate	0 (energyPoint)

FIG. 1 – Tableau des paramètres

La valeur par défaut peut être facilement changée. Pour tester l'effet d'une nouvelle valeur, un tableau présentant tous les paramètres du modèle permet d'en modifier temporairement les valeurs. Par exemple, il est facile de changer le nombre initial de ruminants restreints comme indiqué dans la figure 1. Alors, pour analyser l'effet de valeurs distinctes sur les simulations, une copie des courbes précédentes peut être affichée et comparée avec les nouvelles

courbes de sortie.

#### 4.2 Le multi-fenêtrage

Comme Cormas s'appuie sur l'architecture MVC, il est possible de visualiser l'environnement spatial à travers plusieurs fenêtres. Parce que le modèle est indépendant de la façon de le voir, on peut sélectionner différents points de vue (PoV) pour afficher (ou non) les entités. Dans le menu PoV d'une grille spatiale, des PoVs sont disponibles pour chaque classe du modèle. Par défaut, trois PoVs sont proposés : «*nil*» qui n'affiche pas les instances de la classe, «*defaultPoV*» pour lequel une figure ou une couleur standard est affichée et «*povId*» qui affiche chaque entité avec une couleur différente. Mais il est facile de déterminer d'autres PoVs grâce à l'interface PoVSetter qui permet d'importer des images ou de dessiner des figures correspondant à différents états d'un agent.

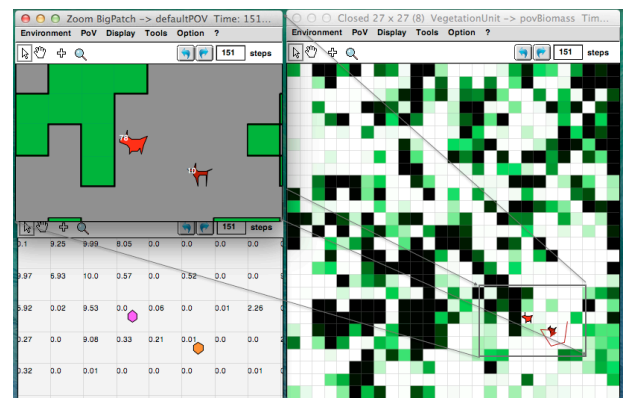


FIG. 2 – Grille spatiale complète (à droite) et deux zooms avec différents points de vue (à gauche).

Un agent est traqué sur la grille de droite ; leur énergie est affichée sur la grille en haut à gauche.

Il est également possible d'ouvrir une autre grille spatiale et sélectionner d'autres PoVs afin d'avoir deux affichages différents simultanément. L'outil "zoom" permet d'ouvrir d'autres fenêtres affichant juste une partie de la grille (sélectionnée par l'utilisateur). Certaines informations peuvent être affichées (par exemple, la biomasse de chaque cellule) et en utilisant un menu contextuel sur un agent, celui-ci peut être tracé. La capture d'écran ci-dessous montre une vue de la grille spatiale complète (partie gauche) et deux zooms sur la même région visualisées avec différents points de vue (partie droite) d'un modèle simple. Appelé "ECEC", il a été conçu par [9] et a été répliqué dans Cormas.

### 4.3 Manipulation des agents

Pour interagir avec une simulation, il est possible de modifier la valeur des paramètres.

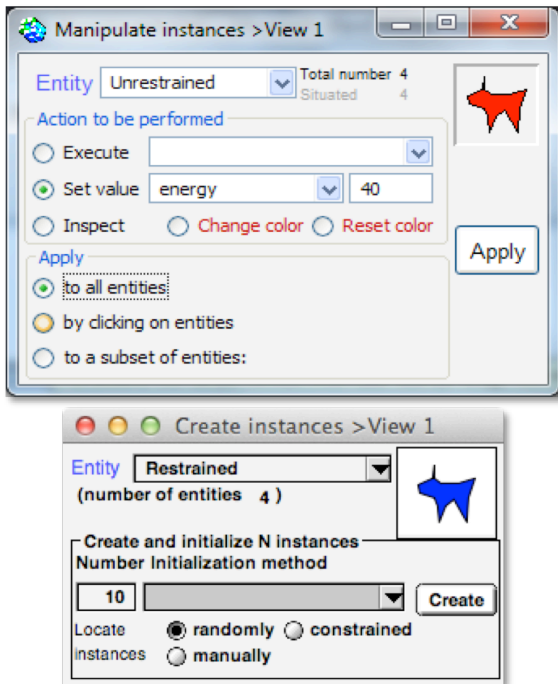


FIG. 3 – Interfaces de manipulation : en bas, pour créer de nouvelles instances ; en haut, pour exécuter une action ou pour changer la valeur des attributs des agents.

Mais il est aussi possible d'agir directement sur l'espace et les agents. Il y a deux façons d'agir : soit sur l'ensemble des entités simultanément, soit sur certaines d'entre-elles. Dans le premier cas, nous pouvons changer l'état d'un groupe d'agents ou en créer de nouveaux (figure 4).

Dans le second cas, l'outil «Manipulation» permet de contrôler un agent individuellement pour le déplacer ou lui envoyer des messages. Un clic droit sur cet agent ouvre un menu contextuel qui permet de sélectionner un message à partir d'une liste contenant les méthodes de cet agent. La figure 4 montre deux façons d'envoyer des messages à un agent : en haut, la méthode "step" sera exécutée par chaque agent lorsque l'utilisateur cliquera dessus, ou en bas, en cliquant sur un agent, puis en sélectionnant "step" depuis le menu déroulant.

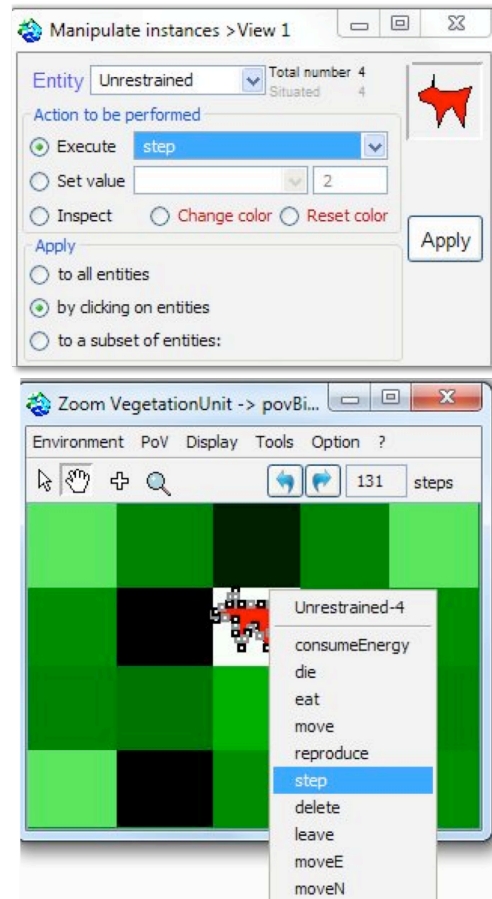


FIG. 4 – Deux façons d'envoyer le message "step" à un agent.

### 4.4 "Habitus": personnalisation des interfaces

Une fonctionnalité permet de personnaliser l'interface spatiale en spécifiant les façons de percevoir et d'interagir avec les entités du modèle. Cette fonctionnalité est basée sur le concept d'*habitus* défini par Bourdieu comme « l'ensemble des manières d'être, de sentir, d'agir et de penser qui sont propres à un individu » [24]. Selon Bourdieu, un *habitus* structure le comportement et les actions d'un individu tout en structurant sa position dans un espace social multidimensionnel.

Lors de l'élaboration d'un SMA hybride, la fonction *Habitus* de Cormas permet au modélisateur de contraindre les différents rôles et perceptions d'un joueur. La définition d'un *habitus* consiste à définir (1) comment l'utilisateur peut voir l'espace : quelles entités sont affichées, quels PoVs et quelles informations sont disponibles, et (2) la façon dont ils peuvent interagir avec cette interface : quelles actions peuvent être exécutées (se

déplacer, consommer, etc.) et sur quelles entités. Ainsi, deux configurations distinctes autorisent des points de vue et des contrôles différents et entraînent des asymétries d'information et d'action entre les joueurs (voir partie 5.3).

#### 4.5 Distribuer les vues pour la simulation interactive

Dans Cormas, une simulation peut être distribuée sur plusieurs machines. Cela signifie que plusieurs utilisateurs peuvent visualiser à distance la même simulation (selon différents points de vue) et manipuler les entités.

L'architecture de distribution ne se conforme pas à la norme IEEE des jeux vidéo. Cette norme établie par l'industrie du jeu commercial et militaire est dédiée aux applications temps réel. Or la simulation massivement en ligne n'est pas notre objectif.

Dans Cormas, la simulation n'est pas dupliquée sur chaque ordinateur, mais seulement les vues et les contrôleurs (selon MVC) sont distribués. Ainsi, un seul serveur héberge la simulation et les machines clientes affichent des points de vue spécifiques et proposent un contrôle des entités. Plusieurs utilisateurs peuvent alors manipuler leurs agents et agir collectivement sur un même environnement virtuel.

Si une simulation peut être distribuée sur Internet, nous préférons utiliser cette capacité sur des ordinateurs en réseau dans la même pièce. Selon nous, la proximité physique est importante car elle permet une interaction plus riche et naturelle par le dialogue direct ou par la communication non verbale.

#### 4.6 Diagramme d'activités exécutable

Dans Cormas, un éditeur permet de dessiner des diagrammes d'activités simples. Lors d'une simulation, ces schémas sont interprétés directement par les agents "à la volée", sans qu'il soit nécessaire de les traduire en code. Il est ainsi possible de modifier le schéma de comportement d'un agent sans avoir à le coder. On peut aussi modifier le simulateur en cours d'exécution, sans arrêter ni redémarrer la simulation.

Par souci de simplicité, les éléments disponibles sur l'éditeur sont réduits : nœud initial et final, points de décision, activités

simples (sans paramètre d'entrée ni de sortie). En sélectionnant une activité ou un point de décision depuis la barre d'outils, l'utilisateur peut ajouter un nouvel élément sur le diagramme. Ensuite, il doit choisir l'opération à effectuer par cet élément : il la sélectionne dans une liste d'activités automatiquement générée à partir des opérations disponibles de la classe cible. Puis l'utilisateur peut dessiner une transition entre deux nœuds. A partir d'un point de décision, il doit dessiner deux transitions : une pour laquelle la réponse au test est vraie (en vert) et une pour un test négatif (en rouge). Ainsi, à partir des opérations de base déjà définies par le modélisateur, on peut générer un nouveau comportement de niveau supérieur, sans aucune connaissance en programmation.

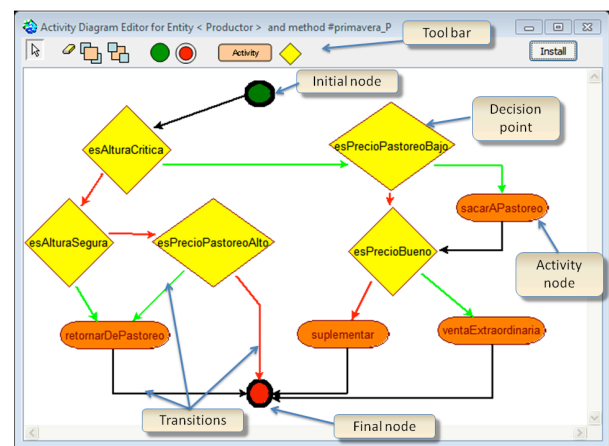


FIG. 5 – L'éditeur de diagramme d'activité

Cet éditeur ne permet pas d'éviter au modélisateur de programmer son SMA. L'objectif est plutôt de concevoir collectivement le comportement d'un agent en organisant la succession des activités par *plug-and-play*. Ces activités de base sont des méthodes ayant été préalablement codées par le modélisateur.

Parce qu'il est destiné aux non-spécialistes, cet éditeur a été conçu pour être aussi simple que possible. Pour cette raison, il ne contient pas de fonctions sophistiquées telles que les *Swimlanes*, les itérations ou les événements simultanés qui sont spécifiés par UML 2.0 [25]. En retour, cette simplicité permet à quiconque de participer plus activement à la conception du modèle avec une plus grande efficacité grâce à l'évaluation immédiate de toute modification.

L'éditeur n'obéit pas à la norme "Executable UML" (xML) [26], [27], dont les

spécifications requièrent un compilateur pour traduire un diagramme en code. Dans Cormas, un diagramme d'activité est directement interprété par les agents. En d'autres termes, chaque diagramme est enregistré dans le modèle au même titre que le reste du code source. Il peut être ré-ouvert à tout moment, modifié et exécuté sans compilation. En tirant parti des avantages de Smalltalk (un langage de programmation réflexif), il est possible de modifier le diagramme d'un agent alors que la simulation est en cours d'exécution. Dès que celui-ci est enregistré, l'agent commence à exécuter son nouveau comportement. Cette spécificité peut être utile lorsqu'un utilisateur qui observe la tendance d'une simulation, veut tester un changement de comportement pour infléchir le cours de la simulation.

#### 4.7 Le voyage dans le temps

A l'instar de nombreuses plateformes, le temps avance par pas-de-temps dans Cormas. Depuis peu, il est possible de remonter le temps d'une simulation. Comme le calcul inverse des activités est mathématiquement impossible, Cormas ne simule pas le recul dans le temps. Pour activer la fonction de retour arrière, les états successifs du système doivent avoir été enregistrés. Alors, un clic sur le bouton « backward » déclenche simplement la restauration de l'état stocké précédent. Ainsi, on peut aller, en avant ou en arrière, à un instant particulier dans le temps de la simulation en rechargeant un état enregistré.

Cette navigation dans le temps permet d'analyser le modèle et de vérifier la cohérence des mécanismes. En cherchant à comprendre un comportement qu'il considère étrange, l'utilisateur peut revenir à un moment précis, juste avant la période anormale. Comme pour un film, il re-visionne le déroulement pour comprendre comment et pourquoi les entités agissent de telle manière. Mais à partir de cet état particulier, il peut aussi relancer une nouvelle simulation pour vérifier si les agents se comportent de façon similaire ou si le système évolue différemment (bifurcation, cf. figure 6).

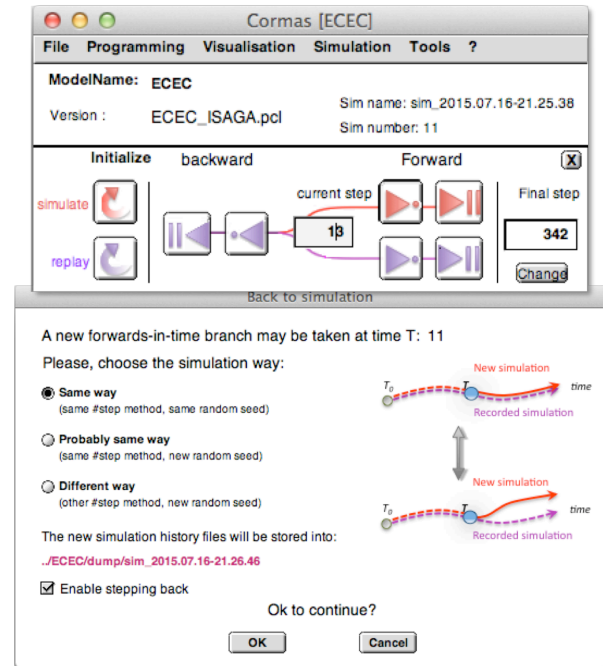


FIG. 6 – En haut: l'interface principale de Cormas avec les boutons 'Simulate' (rouges) et 'Replay' (violets). En bas: l'interface de bifurcation pour recommencer la simulation à partir d'un état enregistré.

Par ailleurs, parce que l'initialisation standard d'une simulation peut créer des artefacts (tous les agents ayant le même âge, par exemple), il est pratique de faire tourner une simulation jusqu'à un état stabilisé. Cet état peut alors être enregistré (*snapshot*) pour devenir le point de départ des simulations futures.

Cette capacité de sauver et de restaurer l'état du système est utilisée lors de la manipulation des agents : les boutons « undo » et « redo » de la grille spatiale permettent d'annuler une action de l'utilisateur ou de la réactiver.

## 5 Mises en pratique

### 5.1 Co-conception d'un SMA sur la viande de brousse avec les villageois d'un parc naturel au Cameroun

Ce projet de développement cherche à aider les populations locales à gérer la faune sauvage autour du parc national de Korup au Cameroun. Il vise à maintenir les animaux surexploités chassés pour la consommation personnelle et pour de l'argent. A cet effet, une utilisation précoce et interactive d'un SMA stylisé a été réalisée avec des chasseurs dans les villages autour du parc [28].

L'un des défis pour concevoir ce modèle était la formulation d'algorithmes de prise de décision pour l'activité de chasse. Deux séries d'ateliers ont été organisées dans les villages de la zone d'étude. Les premiers cherchaient à introduire un module individu-centré spatialisé décrivant la biologie de l'espèce chassée : le céphalophe bleu, une petite antilope forestière. Ensuite, il a fallu préciser les pratiques de chasse des participants grâce à des discussions collectives autour du simulateur. Un exercice en 3 étapes a été mené afin d'impliquer les participants dans la conception du modèle.

La première étape a consisté à dessiner sur la grille spatiale un village abstrait dans la forêt, puis y intégrer le module de dynamique de population du céphalophe. Les différents types de couverts et la notion de cellule de 1 ha ont donc été présentés, ainsi que les 7 couleurs correspondant aux différents stades évolutifs du céphalophe. Dans une deuxième étape, la disposition des pièges dans la forêt a été spécifiée par les chasseurs eux-mêmes, puis traduite en algorithme. Sur cette base, une dernière étape a consisté à représenter les 7 villages de la zone et le parc national. Ainsi, l'espace représenté a été progressivement élargi, de 1,5x1,5 km<sup>2</sup> à 5x5 km<sup>2</sup>, jusqu'à 16x18 km<sup>2</sup>. Ce processus de zoom arrière a permis de commencer en se concentrant sur la biologie et le comportement du céphalophe, puis en introduisant la chasse. L'objectif était de communiquer et de discuter des paramètres et des hypothèses pour que les participants comprennent le modèle et se l'approprient. Dans les derniers ateliers, une représentation plus réaliste de la région a généré des discussions plus concrètes. Pendant les différentes étapes, les fonctionnalités de Cormas ont été largement utilisées pour dessiner collectivement l'espace, modifier interactivement les attributs des entités et pour créer et manipuler directement les entités (les animaux et la pose des pièges). S'en sont suivis des débats collectifs.

## **5.2 Conception collective d'un SMA en utilisant xUML avec des éleveurs en Uruguay**

L'élevage bovin joue un rôle central dans l'économie de l'Uruguay. En raison des graves sécheresses des dernières décennies ayant provoqué la mort de milliers d'animaux et de nombreuses faillites, un projet visant à

améliorer les capacités d'adaptation des éleveurs a été lancé.

Pour tester plusieurs stratégies de gestion et faciliter la communication entre éleveurs et services d'appui, un SMA a été élaboré. La première étape était plus classique puisqu'elle a consisté à concevoir avec des experts un SMA qui décrit la croissance des pâturages, la dynamique du troupeau et des agents imitant grossièrement des stratégies traditionnelles [29]. Cette première version a été présentée et discutée avec les éleveurs bovins de la zone au cours de plusieurs ateliers. Leur principale critique portait sur les comportements trop simplistes de l'agent.

La deuxième étape a consisté à modifier et évaluer le modèle avec les éleveurs. Afin de rendre cette évaluation plus vivante et efficace, nous avons conçu l'outil xUML présenté dans la section 4.6. A partir des activités de base déjà disponibles, les participants étaient capables de générer de nouvelles stratégies de gestion, sans aucune connaissance en programmation.

L'utilisation de l'éditeur xUML a révélé deux caractéristiques intéressantes. Tout d'abord, en étant capable de modifier le comportement des agents, les participants ont joué avec le modèle et ont mieux compris sa logique. La réponse immédiate du modèle à chaque modification a augmenté leur compréhension des mécanismes sous-jacents. Des débats sur la meilleure façon de faire face aux sécheresses s'en sont suivis. En conclusion, bien que les stratégies de gestion proposées semblaient trop simplistes, de nombreux éleveurs se sont finalement reconnus comme producteurs traditionnels.

La deuxième caractéristique concerne des aspects techniques du modèle. En testant des stratégies alternatives avec l'éditeur xUML, les participants ont identifié certains biais : ils ont réalisé qu'en cas de sécheresse, les agents réagissaient toujours trop tard et la décision par exemple de nourrir le troupeau avec des suppléments n'empêchait apparemment pas la crise. Les participants ont compris que les agents doivent agir plus fréquemment qu'une seule fois par saison. Le modèle a été corrigé en conséquence en répétant les activités de l'agent chaque semaine.

Les résultats de ces exercices collectifs ont dépassé nos attentes. Au-delà des débats qu'il a suscités, le SMA a contribué à identifier des



stratégies d'adaptation qui semblent améliorer la résilience des producteurs. Aujourd'hui, la plupart des agriculteurs et des techniciens ayant participé aux ateliers, poursuivent l'expérience avec le modèle. Ils l'utilisent pour rechercher des stratégies de gestion plus efficaces dans les périodes normales et de sécheresse. Nous espérons que cela facilitera l'émergence de pratiques résilientes face au changement climatique [30].

### 5.3 Sensibiliser les acteurs aux multiples points de vue par des simulations asymétriques distribuées

En France, l'étalement urbain entraîne des pertes de surfaces agricoles et forestières et la destruction des habitats pour la biodiversité. En 2010, l'engagement du "Grenelle 2" vise à réduire ces impacts par la planification urbaine. Pourtant, au niveau local, ces mesures font face à des difficultés liées à la répartition des tâches et des responsabilités concernant la gestion de l'environnement. Les acteurs locaux manquent d'une vue d'ensemble et ne comprennent pas toujours pourquoi et comment les différents acteurs s'engagent à conserver la biodiversité.

Pour répondre à cette question, un SMA a été développé qui simule les interactions entre le changement d'utilisation des terres et deux espèces emblématiques : le busard cendré (classé sur la liste rouge des oiseaux de proie migrateurs qui jouent un rôle dans la régulation des ravageurs et la dispersion des graines), et les abeilles de Montagu (qui fournissent des services écosystémiques aux forêts de châtaigniers). La qualité des eaux souterraines qui résulte du type d'utilisation des terres, est aussi modélisée. Le SMA décrit enfin les actions de cinq acteurs principaux : un maire, un promoteur immobilier, un forestier, un agriculteur et un écologiste.

Chacun d'eux a un objectif spécifique lié à l'utilisation des terres [31]. Ces acteurs sont représentés par des avatars dont les décisions sont prises par les joueurs. Ainsi, ce modèle hybride mélange des processus informatisés (dynamique écologique, demande du marché foncier, popularité des élus,...) et des décisions humaines (achat de terres, choix de pratiques agricoles, récolte du bois de construction, délivrance de permis de construire, promotion de pratiques respectueuses de l'environnement). Lors d'un

atelier, la simulation est distribuée sur plusieurs machines : une pour chaque rôle. Sur chacune, un point de vue spécifique sur l'environnement est affiché et un nombre limité d'actions est disponible (figure 7). Cette personnalisation utilise l'interface Habitus.

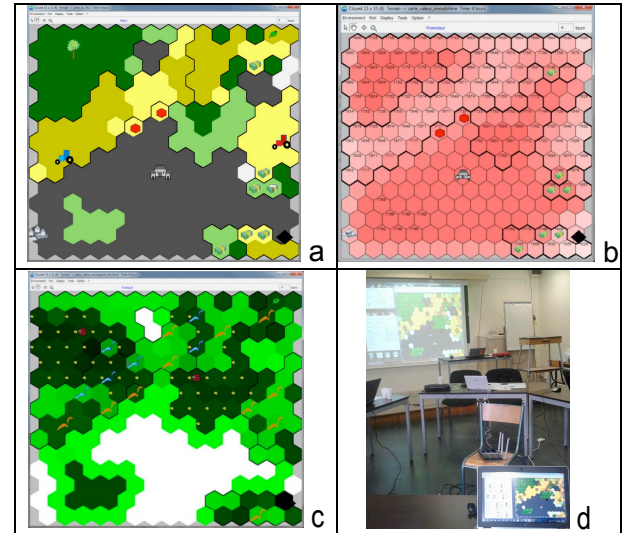


FIG. 7 – (a) l'habitus du maire affiche le plan de développement urbain qui permet au joueur d'agir sur les transactions foncières symbolisées par des billets de banque. (b) La carte du promoteur représente les prix du foncier. (c) l'habitus de l'écologiste fournit des informations sur l'habitat et montre l'emplacement et la santé des busards et des abeilles de Montagu. (d) un atelier de jeu

Une telle asymétrie de visualisation et d'action permet aux participants de se plonger dans le rôle des véritables acteurs et de mieux comprendre leurs contraintes, leurs points de vue et les raisons de leurs choix souvent mal compris par les autres et qui pénalisent souvent la biodiversité au final.

## 6 Perspectives et conclusions

Cormas est principalement destiné aux scientifiques non informaticiens. L'objectif est de faciliter la conception, l'implémentation et l'évaluation de modèles fiables et efficaces. Pour cela, Cormas doit rester aussi simple que possible. C'est la raison pour laquelle Cormas ne propose pas de système continu (ni temporel, ni spatialement). Le formalisme DEVS [32], par exemple, est plus sophistiqué et permet de simuler des mouvements réalistes comme les collisions. Mais dans le domaine de la gestion des ressources, ces raffinements ne semblent pas essentiels. Des concepts plus grossiers

paraissent suffisants pour résoudre les problèmes de ce domaine et pour lequel la question repose davantage sur une compréhension globale du système plutôt que sur la précision temporelle des interactions.

Pour aider les thématiciens à implémenter leurs modèles sans dépendre d'informaticiens, les développements de Cormas se concentrent sur la conception d'interfaces utilisateur pour générer automatiquement une partie du code informatique. Un éditeur de diagramme de classes, qui générera le code des classes, attributs et associations est prévu pour 2016.

Nous développons également un éditeur plus complet de diagramme d'activité permettant de manipuler les variables. Le modélisateur définira des points de décision en décrivant graphiquement le test à partir des paramètres du modèle et d'opérateurs logiques. Ces diagrammes d'activité avec paramètres d'entrée et de sortie seront utiles pour définir des opérations plus complexes. Cependant cet outil sera difficilement utilisable avec des acteurs non scientifiques.

Une connexion avec R, logiciel de calcul statistique permettra d'exécuter des analyses de sensibilité complexes à partir de R et d'obtenir des graphiques statistiques de qualité.

Mais l'orientation principale de Cormas reste l'interactivité avec les acteurs du développement. C'est la raison pour laquelle le principal effort est consacré aux interfaces homme-machine et au design ergonomique. Par exemple, nous développons actuellement une extension pour contrôler le mouvement des agents sur la grille spatiale à travers des objets réels qui sont physiquement déplacés sur une table. Ce développement utilise la reconnaissance numérique de QR codes imprimés sur le dessus des objets physiques. En projetant la grille spatiale sur une table, cette extension sera utilisée dans des simulations hybrides mêlant environnements virtuels et objets réels. Tous ces développements ont lieu en parallèle avec des expériences concrètes sur le terrain engageant divers acteurs locaux afin d'apporter des contributions significatives sur les questions sociales et environnementales.

## Références

[1] F. Bousquet, I. Bakam, H. Proton, et C. L. Page, « Cormas: Common-pool resources and multi-agent systems », in *Tasks and Methods in Applied Artificial*

*Intelligence*, A. P. del Pobil, J. Mira, et M. Ali, Éd. Springer Berlin Heidelberg, 1998, p. 826–837.

[2] C. Le Page, N. Becu, P. Bommel, et F. Bousquet, « Participatory Agent-Based Simulation for Renewable Resource Management: The Role of the Cormas Simulation Platform to Nurture a Community of Practice. », *Journal of Artificial Societies & Social Simulation*, vvol. 15, no. 1.

<<http://jasss.soc.surrey.ac.uk/15/1/1.html>>, 2012.

[3] U. Wilensky et K. Reisman, « Thinking like a Wolf, a Sheep, or a Firefly: Learning Biology through Constructing and Testing Computational Theories-An Embodied Modeling Approach », *Cognition and Instruction*, vol. 24, n° 2, p. 171–209, janv. 2006.

[4] O. Barreteau et others, « Our Companion Modelling Approach », *Journal of Artificial Societies and Social Simulation* vol. 6, no. 1.

<<http://jasss.soc.surrey.ac.uk/6/2/1.html>>, 2003.

[5] M. Étienne, Éd., *Companion Modelling. A Participatory Approach to Support Sustainable Development*. Dordrecht: Springer Netherlands, 2011.

[6] C. Le Page, G. Abrami, O. Barreteau, N. Becu, P. Bommel, A. Botta, A. Dray, C. Monteil, et V. Souchère, « Models for sharing representations », in *Companion Modelling*, Springer Netherlands, 2014, p. 69–101.

[7] A. Voinov et F. Bousquet, « Modelling with stakeholders », *Environmental Modelling & Software*, vol. 25, n° 11, p. 1268–1281, nov. 2010.

[8] J. Brauer, « The VisualWorks Development Environment », in *Programming Smalltalk – Object-Oriented from the Beginning*, Springer Fachmedien Wiesbaden, 2015, p. 77–96.

[9] Pepper J.W. et Smuts B.B., « The evolution of cooperation in an ecological context: an agent-based model », *Dynamics of human and primate societies: agent-based modeling of social and spatial processes*, Oxford, p. 45–76, 2000.

[10] T. Reenskaug, « The Model-View-Controller (MVC) Its Past and Present », *University of Oslo Draft*, 2003.

[11] F. Bousquet, O. Barreteau, C. Mullon, et J. Weber, « Modélisation d'accompagnement: systèmes multi-agents et gestion des ressources renouvelables », présenté à Quel environnement au XXIème siècle ? Environnement, maîtrise du long terme et démocratie, Abbaye de Fontevraud, France, 1996.

[12] [Collectif ComMod] F. Bousquet, M. Antona, S. Aubert, C. Barnaud, O. Barreteau, N. Becu, S. Boisseau, P. Bommel, A. Botta, J. Castella, et others, *La posture d'accompagnement des processus de prise de décision: les références et les questions transdisciplinaires*, in *Modélisation de l'environnement : entre natures et sociétés.*, Quae. Hervé Dominique (ed.), Laloë Francis (ed.), 2009, p. 71–89.

- [13] T. R. Gurung, F. Bousquet, et G. Trébuil, « Companion modeling, conflict resolution, and institution building: sharing irrigation water in the Lingmteychu Watershed, Bhutan », *Ecology and Society* 11(2): 36, p. 36, 2006.
- [14] F. Bousquet, *Companion Modeling and Multi-agent Systems for Integrated Natural Resource Management in Asia*. Int. Rice Res. Inst., 2005.
- [15] E. von Glasersfeld, « Le Moigne's defense of constructivism », *GRASCE, Entre systémique et complexité, chemin faisant [Between systemics and complexity – making the way]*. Available at <http://www.vonglasersfeld.com/225>, Paris, p. 85–90, 1999.
- [16] J.-L. Le Moigne, 1995. *Les épistémologies constructivistes*, Presses Universitaires de France.
- [17] C. Le Page, D. Bazile, N. Becu, P. Bommel, F. Bousquet, M. Etienne, R. Mathevet, V. Souchère, G. Trébuil, et J. Weber, « Agent-Based Modelling and Simulation Applied to Environmental Management », in *Simulating Social Complexity*, Springer Berlin Heidelberg, 2013, p. 499–540.
- [18] W. Daré et O. Barreteau, « A role-playing game in irrigated system negotiation: between play and reality. », *Journal of Artificial Societies and Social Simulation*, vol. 6, n° 3, <<http://jasss.soc.surrey.ac.uk/6/3/6.html>>, 2003.
- [19] O. Barreteau, « The joint use of role-playing games and models regarding negotiation processes: characterization of associations », *Journal of Artificial Societies and Social Simulation* vol. 6, no. 2. <<http://jasss.soc.surrey.ac.uk/6/2/3.html>>, 2003.
- [20] F. Bousquet, O. Barreteau, P. d' Aquino, M. Etienne, S. Boissau, S. Aubert, C. Le Page, D. Babin, et J.-C. Castella, « Multi-agent systems and role games: collective learning processes for ecosystem management », *Complexity and Ecosystem Management. The Theory and Practice of Multi-Agent Systems*, Edward Elgar, Londres, p. 248–286, 2002.
- [21] M. A. Janssen et E. Ostrom, « Empirically based, agent-based models », *Ecology and Society*, vol. 11, n° 2, p. 37, 2006.
- [22] P. D'Aquino, C. Le Page, F. Bousquet and A. Bah « Using Self-Designed Role-Playing Games and a Multi-Agent System to Empower a Local Decision-Making Process for Land Use Management: The SelfCormas Experiment in Senegal », *Journal of Artificial Societies and Social Simulation* <<http://jasss.soc.surrey.ac.uk/6/3/5.html>>, 2003.
- [23] G. Deffuant, G. Weisbuch, F. Amblard, et T. Faure, « Simple is beautiful? and necessary », *Journal of Artificial Societies and Social Simulation*, vol. 6, n° 1, <<http://jasss.soc.surrey.ac.uk/6/1/6.html>>, 2003.
- [24] P. Bourdieu, *Le Sens pratique*, Éditions de Minuit. Paris, 1980.
- [25] OMG, « The Unified Modeling Language Specification v2.0 ». 2005.
- [26] S. J. Mellor et M. Balcer, *Executable UML: A Foundation for Model-Driven Architectures*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.
- [27] OMG, « Model Driven Architecture, MDA Guide Version 1.0.1 », 2008.
- [28] C. Le Page, K. S. Bobo, T. O. W. Kamgaing, B. F. Ngahane, et M. Waltert, « Interactive Simulations with a Stylized Scale Model to Codesign with Villagers an Agent-Based Model of Bushmeat Hunting in the Periphery of Korup National Park (Cameroon) », *Journal of Artificial Societies and Social Simulation*, vol. 18, n° 1, p. 8, 2015.
- [29] F. J. Dieguez Cameroni, R. Terra, S. Tabarez, P. Bommel, J. Corral, D. Bartaburu, M. Pereira, E. Montes, E. Duarte, et H. Morales Grosskopf, « Virtual experiments using a participatory model to explore interactions between climatic variability and management decisions in extensive grazing systems in the basaltic region of Uruguay », *Agricultural Systems*, vol. 130, p. 89–104, sept. 2014.
- [30] P. Bommel, F. Dieguez, D. Bartaburu, E. Duarte, E. Montes, M. Pereira Machin, J. Corral, C. J. P. de Lucena, et H. Morales Grosskopf, « A Further Step Towards Participatory Modelling. Fostering Stakeholder Involvement in Designing Models by Using Executable UML », *Journal of Artificial Societies and Social Simulation*, vol. 17, n° 1, <<http://jasss.soc.surrey.ac.uk/17/1/6.html>>, 2014.
- [31] N. Becu, N. Frascaria-Lacoste, et J. Latune, « Distributed Asymmetric Simulation - Enhancing Participatory Simulation Using the Concept of Habitus. », présenté à ISAGA 2014: The shift from teaching to learning: Individual, Collective and Organizational Learning through Gaming Simulation, Dornbirn, Austria, Bertelsmann, 2014.
- [32] B. P. Zeigler, H. Praehofer, et T. G. Kim, *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, 2000.